CS 744: PIPEDREAM

Shivaram Venkataraman

Fall 2022

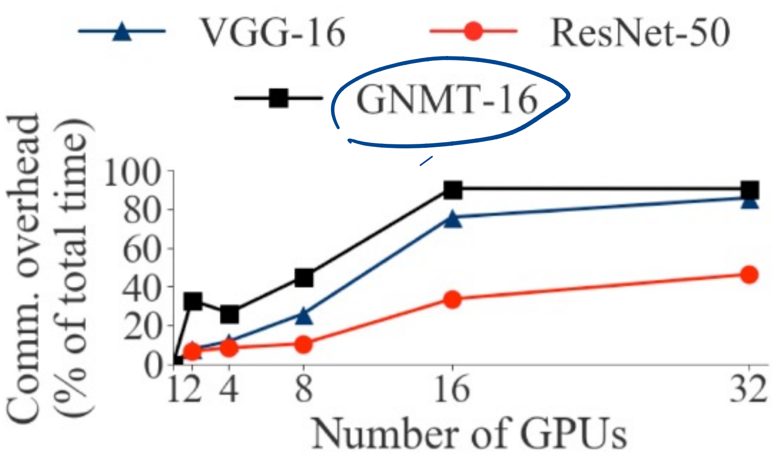
ADMINISTRIVIA

- Assignment 2 is due Wednesday AM! → Please post on Piazza

- Course project preference sheet: Out today!
  - Propose your own?
  - Or rank 1 through 5 of some project ideas we have
  - Group!

LIMITATIONS OF DATA PARALLEL

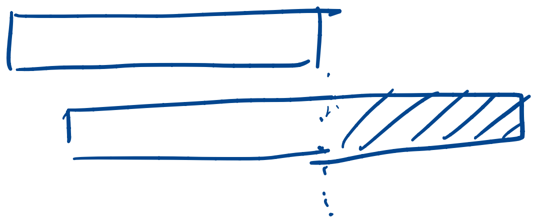
- Overhead increases as num GPUs increases



8xV100s with NVLink (AWS)
PyTorch + NCCL 2.4

5

-



backward pass

AllReduce

very high for some model types

- Overhead is lower <= 8 GPUs, higher after that

1 to 8 GPUs

machine

8 GPUs

machine

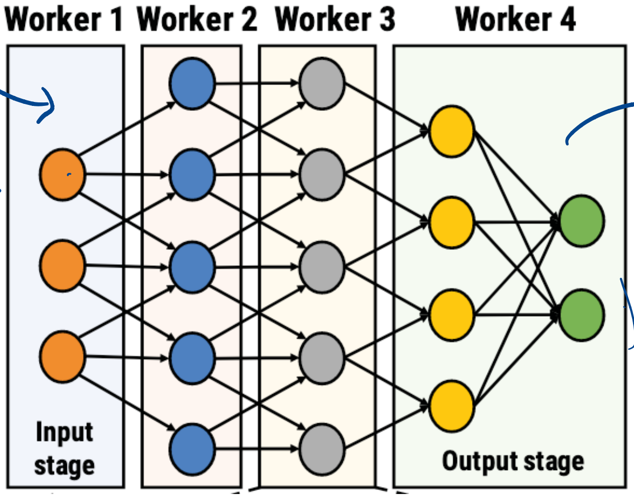
"fraction of training time spent in communication stalls"

MODEL PARALLEL TRAINING

5 layers in model

stage 1

input



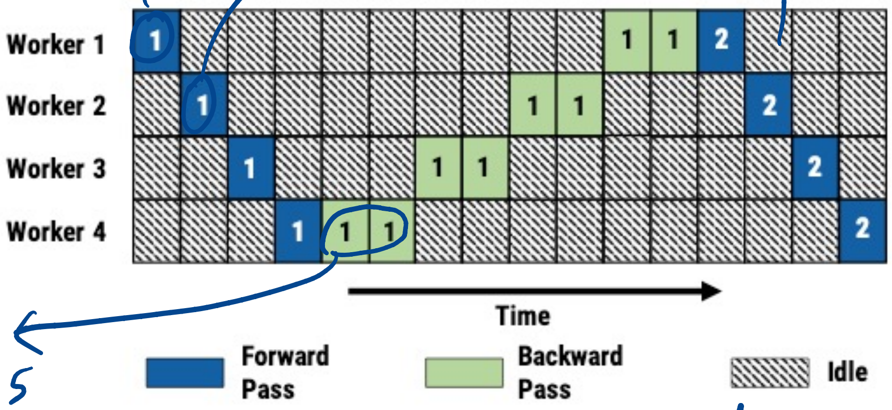
stage 4

gradients here

forward pass layer 1

fwd pass layer 2

low utilization



backward pass for layer 4,5

1. Comm is constant as num workers increases. limited connection
2. Memory req. are lower & can use heterogeneous machines.

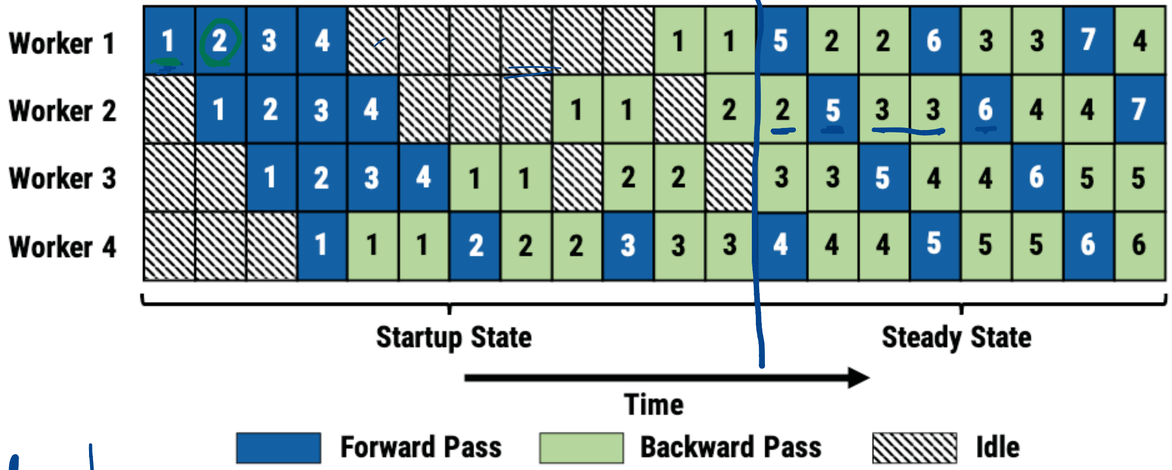
PIPELINE PARALLEL

Key Idea:
Instead of 1 input batch
feed in "k" batches

- Handle scenarios where some stages are slow.

k = 4 here

steady state util is high



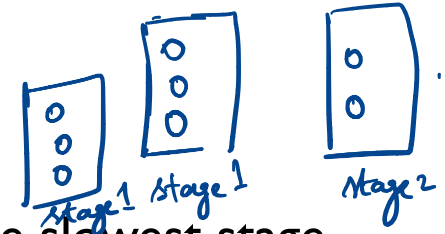
Advantages?

- fixed comm. overhead
- high utilization
-

- Partitioning
- Scheduling of FWD, BCKWD
- Learning

CHALLENGE 1: WORK PARTITIONING

similar time?



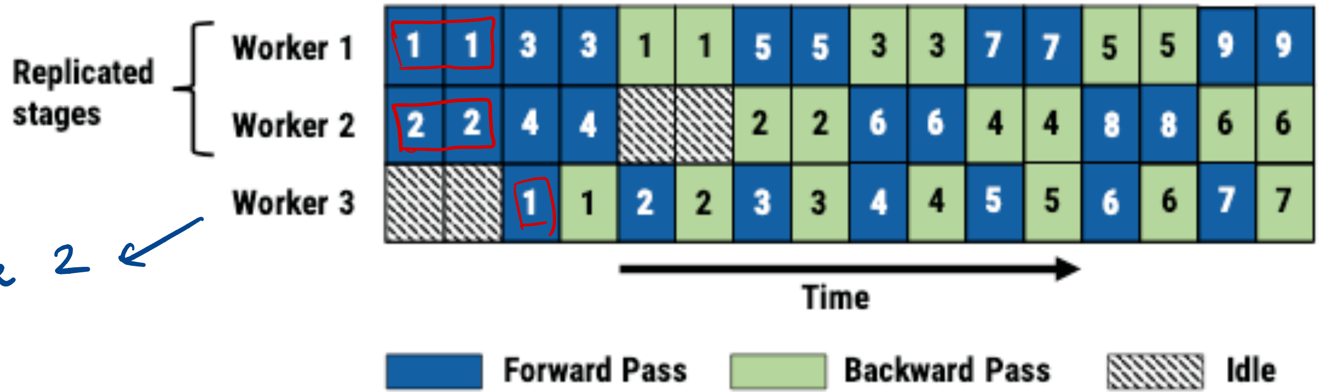
Goal: Balanced stages in the pipeline. Why?

Steady state throughput is the throughput of the slowest stage

Hybrid parallel

Stages can be replicated! Ex: Two stage pipeline, but first stage is replicated

Doing stage 1



stage 2

WORK PARITIONING

Profiler: computation time for forward, backward for each layer

size of output activations, gradients (network transfer)

size of parameters (memory)

Given

- model

- cluster

↳ GPUs, network

Dynamic programming algorithm

Intuition: Find optimal partitions within a server,

Then find best split across servers using that

Static planning

CHALLENGE 2: WORK SCHEDULING

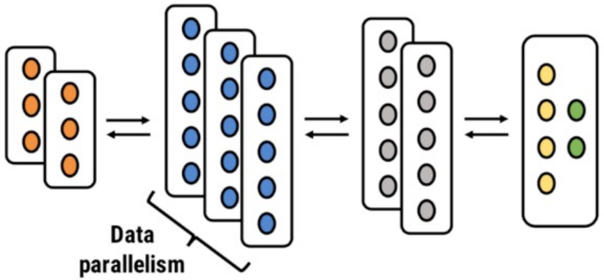
Traditional data parallel

forward iter(i) Iteration 0

backward iter(i)

forward iter(i+1) Iteration 1

…



Pipeline parallel: Worker can

Forward pass to push to downstream

Backward pass to push to upstream

Either run fwd pass next batch or backward pass for prior batch

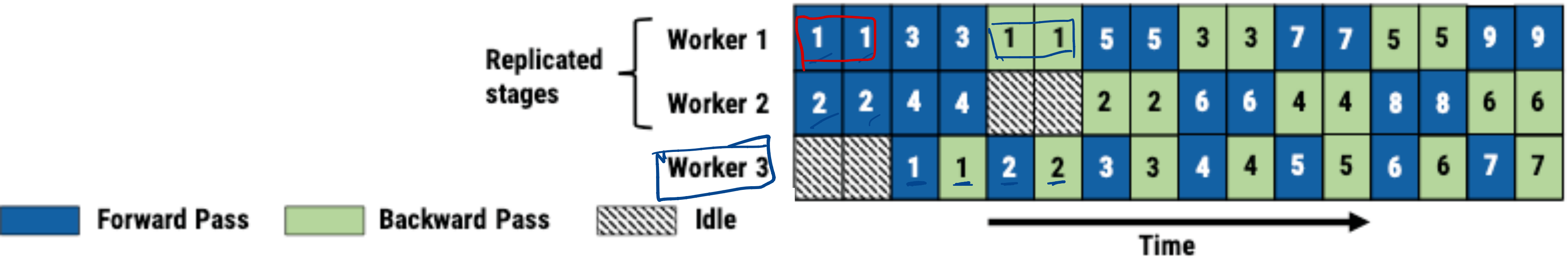
CHALLENGE 2: WORK SCHEDULING

Num active batches ~= num_workers / num_replicas_input

Schedule one-forward-one-backward (1F1B) – Worker 3 → makes sure pipeline is making progress

Round-robin for replicated stages → Worker 2

same worker for fwd, backward

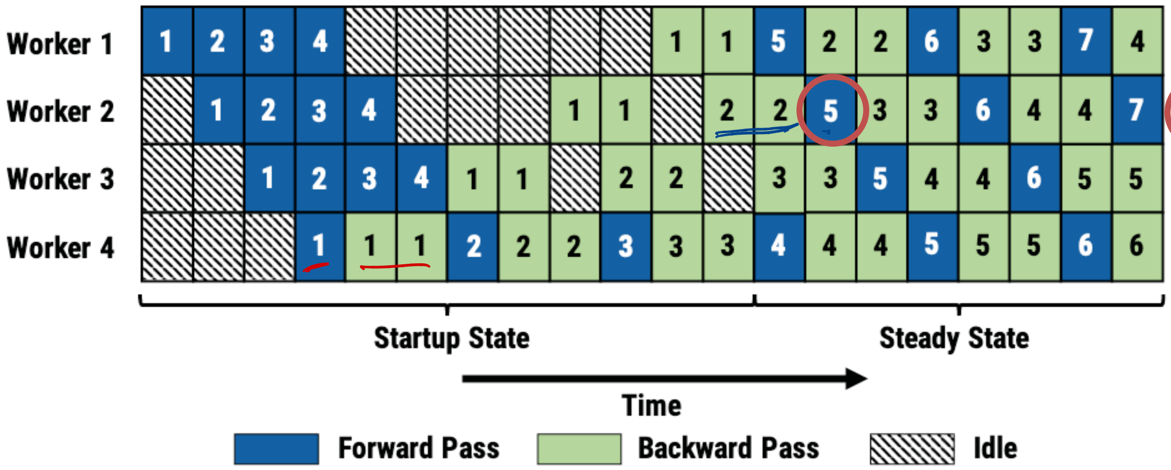


CHALLENGE 3: EFFECTIVE LEARNING

Naïve pipelining

Different model versions forward and backward

Batch no. 5
↳ fwd pass uses model v2.
↳ bck pass used model v4.



5

Keep model v2 stashed

batch 5 ≡ v2

CHALLENGE 3: EFFECTIVE LEARNING

batch no. 5 = [32 images] → $\underbrace{m_{v_1}}_{W1}$ → Activations → $\underbrace{m_{v_2}}_{v_2}$ → Acti

Weight stashing

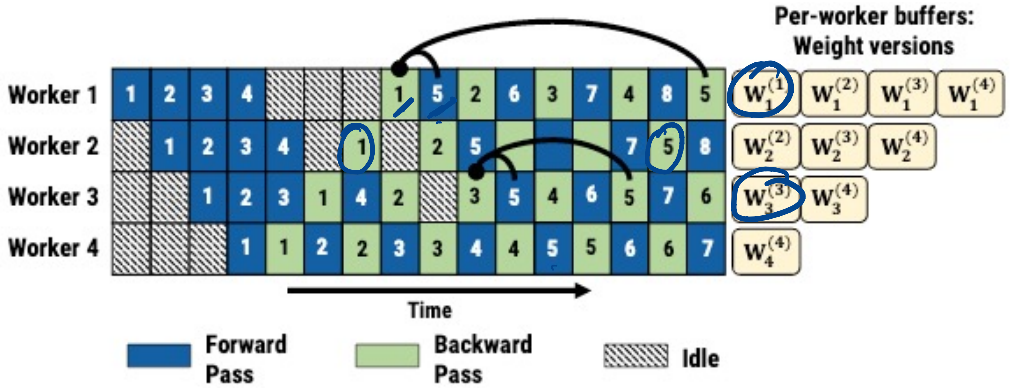
- Maintain multiple versions of the weights
- One per active mini-batch

Use latest version for forward pass.

Retrieve for backward

No guarantees across stages!

↳ diff model versions used in diff stages



STALENESS, MEMORY OVERHEAD

How to avoid staleness:

Vertical sync → in the first stage, record model version used

propagate that across workers

Memory overhead

Similar to data parallel?

↳ lose this benefit with vertical sync.

SUMMARY

Pipeline parallelism: Combine inter-batch and intra-batch

Partitioning: Replication, dynamic programming

Scheduling: 1F1B

Weight management: Stashing, vertical sync

DISCUSSION

https://forms.gle/5cf16BWn6Dziey6e6

Data parallel - not always bad

- speedup higher VGG-16

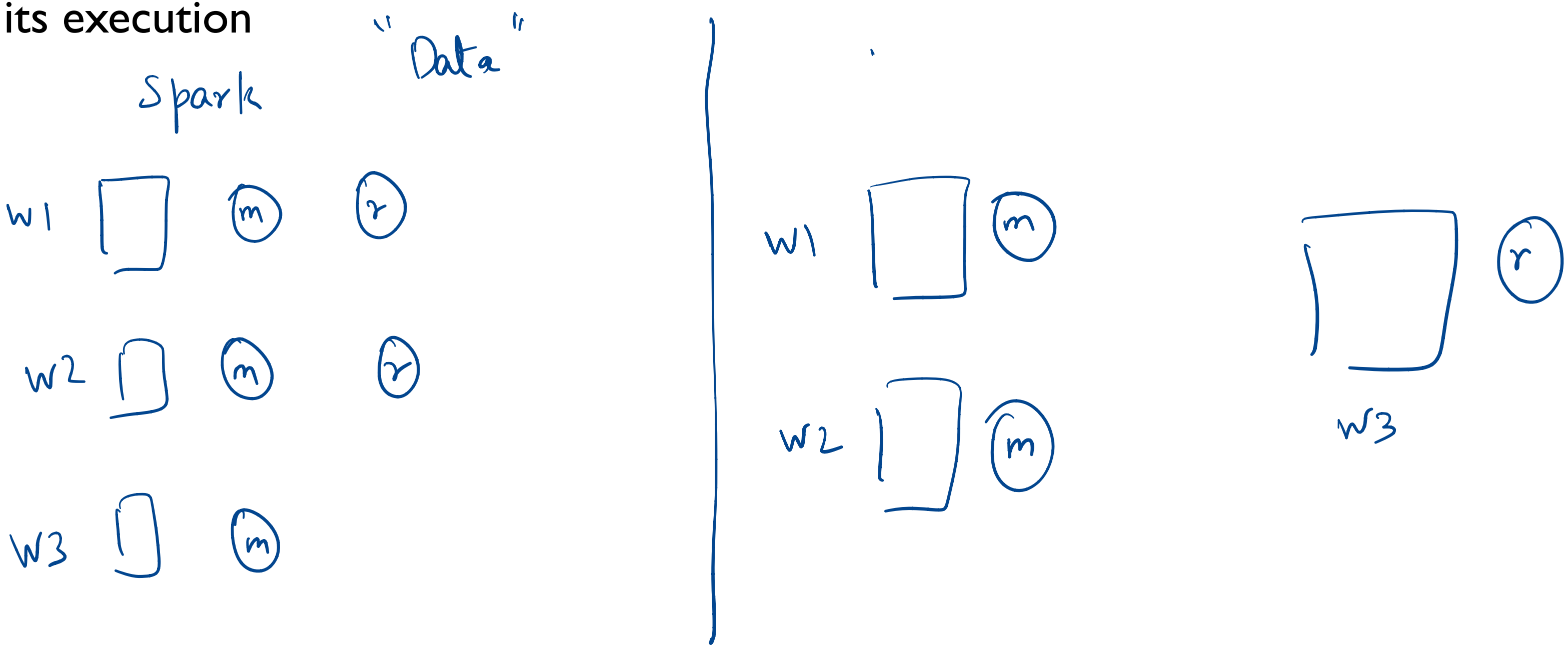
4x4 → more overhead for data parallel

List two takeaways from the following table

Model Name	Model Size	GPUs (#Servers x #GPUs/Server)	PipeDream Config	Speedup over DataParallel (Epoch Time)
Resnet-50	97MB	4x4 2x8	16 16	1x 1x
VGG-16	528MB	4x4 2x8	15-1 15-1	5.28x 2.98x
GNMT-8	1.1GB	3x4 2x8	Straight 16	2.95x 1x

Why ??

What are some other workload scenarios (e.g. things we discussed for MapReduce or Spark) that could use similar ideas of pipelined parallelism? Develop such one example and its execution



NEXT STEPS

Next class: Parameter Server

Assignment 2 is due soon!

Course project preference form out today!